

**SERVER SIDE PROGRAM INTERFACE TO SERVICE
LOGIC EXECUTION ENVIRONMENT**

Inventors:

Thomas E. Creamer

Zygmunt A. Lozinski

Victor S. Moore

Glen R. Walters

International Business Machines Corporation

IBM DOCKET NO. BOC9-2000-0091

IBM DISCLOSURE NO. BOC8-2000-0052

EXPRESS MAIL LABEL NO. EK972213778US

BACKGROUND OF THE INVENTIONTechnical Field

This invention relates to the field of telecommunications, and more particularly, to an interface for accessing a service logic execution environment from a server side program.

Description of the Related Art

The development of the open network application programming interface (API) represents an important departure from traditional methods for opening the architecture of the public switched telephone network (PSTN). Presently, the Advanced Intelligent Network (AIN) architecture defines a call model which allows the creation of telecommunications service applications outside of the switch environment. Telecommunications service applications are a la carte telecommunications applications which can perform enhanced services for a telecommunications session established among two or more parties. Exemplary service applications can include Call Waiting, Caller ID, Call Forwarding, Voice Activated Dialing, and Meet-me Conferencing.

When AIN first had been introduced, in terms of the service application creation process, the AIN architecture represented an important advance. AIN separated service development from switching, allowing service logic components to be developed more quickly and placed in specialized network elements attached to databases. Switches, in turn, being free from all service logic, could be optimized for speed and efficiency. Still, typical service applications developed to the AIN specification are written in specialized languages by specially trained programmers using specialized service creation environments.

Importantly, future telecommunications networks will be characterized by new and evolving network architectures where packet-switched, circuit-switched, and wireless networks are integrated to offer subscribers an array of innovative multimedia, multiparty applications. Equally important, it is expected that the process by which

telecommunications applications are developed will change, and will no longer solely be the domain of the telecommunications network or service application provider. In fact, in order to provide a broad portfolio of novel, compelling applications rapidly, service application providers will increasingly turn to third-party applications developers and software vendors. Thus, application development in the telecommunications domain will become more similar to that in software and information technology in general, with customers reaping the benefits of increased competition, reduced time to market, and the rapid leveraging of new technology as it is developed.

To make this vision a reality, the principles of AIN have been discarded in favor of a new service application component development paradigm. Specifically, it has been recognized that future integrated networks must offer application developers a set of standard, open APIs so that applications written for compatibility with one vendor's system can execute in the system of another vendor. In consequence, the cost of applications development can be amortized, reducing the final cost to the customer. Java APIs for Integrated Networks (JAIN) fulfills the requirements of the new service application component development paradigm. Presently, JAIN includes standard, open published Java APIs for next-generation systems consisting of integrated Internet Protocol (IP) or asynchronous transport mode (ATM) networks, PSTN, and wireless networks. The JAIN APIs include interfaces at the protocol level, for different protocols such as Media Gateway Control Protocol (MGCP), Session Initiation Protocol (SIP), and Transactional Capabilities Application Part (TCAP), as well as protocols residing in the higher layers of the telecommunications protocol stack.

JAIN includes a set of integrated network APIs for the Java platform and an environment to build and integrate JAIN components into services or applications that work across PSTN, packet and wireless networks. The JAIN approach integrates wireline, wireless, and packet-based networks by separating service-based logic from network-based logic. Figure 1 illustrates a conventional JAIN implementation. As shown in Figure 1, a conventional JAIN implementation can include a protocol layer 102 which can include interfaces to IP, wireline and wireless signaling protocols. These

protocols can include TCAP, ISUP, INAP, MAP, SIP, MGCP, and H.323. The JAIN implementation also can include a signaling layer 103 which can include interfaces to provide connectivity management and call control. The conventional JAIN implementation also can include an application layer 104 for handling secure network access and other external services. Finally, the conventional JAIN implementation can include a service layer 106 which can include a service creation and carrier grade service logic execution environment (SLEE) 108.

In JAIN, the protocol layer 102 and the signaling layer 103 are based upon a Java standardization of specific signaling protocols and provide standardized protocol interfaces in an object model. Additionally, applications and protocol stacks can be interchanged, all the while providing a high degree of portability to the applications in the application layer using protocol stacks from different sources. By comparison, the application layer 104 provides a single call model across all supported protocols in the protocol layer 102. Fundamentally, the application layer 104 provides a single state machine for multiparty, multimedia, and multiprotocol sessions for service components in the application layer 104. This state machine is accessible by trusted applications that execute in the application layer 104 through a call control API.

Notably, applications or services executing at the service level 102 can communicate directly with protocol adapters in the SLEE 108. Protocol adapters typically are class methods, callbacks, event or interfaces that encapsulate the underlying resources such as TCAP, MGCP, etc. The underlying resources can be implemented in many programming languages, but a JAIN-conformant protocol product must provide at least the relevant JAIN API. In contrast, an external application or service executing in the application layer 104 does not have to be aware of the underlying resources and can remain oblivious to the fact that some of its session or call legs may be using different protocols.

Service components 112 are the core JAIN components and can execute in the SLEE 108. More particularly, service components 112 are constructed according to a standard component model and, instantiations of component assemblies can execute

in coordination with the SLEE 108. Using information regarding the protocol layer 102 which can be incorporated into the SLEE 108, service components 112 can interact with the underlying protocol stacks without having specific knowledge of the protocol stack. Thus, service components 112 can use the call model provided by the signaling
5 layer to implement telephony services. More importantly, the SLEE 108 can relieve the service components 112 of conventional lifecycle responsibilities by providing portable support for transactions, persistence, load balancing, security, and object and connection instance pooling. In this way, the service components 112 can focus on providing telephony services.

10 Despite the advantages afforded by JAIN, presently, communicating with a service component in the SLEE can be problematic for external applications. In particular, an external application must be programmatically bound to those service components in the SLEE with which the external application can communicate. This problem can be compounded where the external application is distributed across a
15 computer communications network such as the Internet. Accordingly, any attempt to integrate services or other functionality available via the Internet, including the World Wide Web, must be implemented using a separate, proprietary system.

SUMMARY OF THE INVENTION

The invention disclosed herein addresses the inherent deficiencies of the prior art by providing an interface to external applications for accessing service components in a service logic execution environment (SLEE). In particular, the interface can be used in coordination with a server-side program or script executing in a network server such as a Web server. In accordance with the inventive arrangements, external applications can communicate with a Web server across a computer communications network such as the Internet. In particular, external applications can interact with server-side programs or scripts, for example servlets, which can establish communications links with a specially configured service component in the SLEE.

The specially configured service component can act as a conduit between other service components in the SLEE and the external application. Specifically, messages received in the specially configured service component can be posted to an event handler in the SLEE which, in turn, can route the posted messages to target service components. Correspondingly, messages from the target service components can be posted to the event handler in the SLEE which in turn can route the messages to the specially configured service component. The specially configured service component, in turn, can forward the messages to the external application via the server-side program. Thus, service components in the SLEE can freely communicate with an external application without having been programmatically bound to the external application.

One aspect of the present invention can include an advanced intelligent network configured in accordance with the present invention. In particular, an advanced intelligent network according to the present invention can include a SLEE. The SLEE can include an event handler for routing messages between and among client components and service components. At least one service component configured to post and receive messages to and from other service components in the SLEE through the event handler can be included. At least one Internet enabled service component (IESC) executing in the SLEE also can be included. The IESC can be configured to

post and receive messages to and from other service components in the SLEE through the event handler. The IESC can be communicatively linked to a server side program external to the SLEE.

Another aspect of the present invention can include an IESC for use with a SLEE in a telephony environment. The IESC can include at least one client service instance. Each client service instance can correspond to an Internet service application. The client service instance can include a content interface for publishing an interface to the client service instance. A service wrapper can be included. The service wrapper can provide a common interface to the at least one client service instance for routing events between the SLEE and the at least one client service instance. The service wrapper can include a service interface for publishing an interface to the service wrapper. The service wrapper also can include a protocol stack for managing communications in the telephony environment and a deployment descriptor for providing configuration information to the SLEE. The IESC can interact with other generic service components, external applications, service components, as well as protocol stacks.

Another aspect of the invention can include a method for providing an external interface to a SLEE comprising an event handler for routing messages between service components executing in the SLEE. The method can include an IESC registering with the SLEE and the IESC receiving a first event routed by the SLEE. The first event can correspond to an Internet service application which the IESC has registered to receive from the SLEE. Also, the first event can be from a protocol stack, a generic service component, a service component, or an external server side program associated with the Internet service application.

The method further can include establishing a communications link between the IESC and the Internet service application. At least one client service for communicating with the Internet service application can be instantiated and can interact with the Internet service application. A second event can be posted to the SLEE responsive to the interaction between the client service instance and the Internet service application. The second event can correspond to the interaction.

Yet another aspect of the invention can include a machine readable storage, having stored thereon a computer program having a plurality of code sections executable by a machine for causing the machine to perform a series of steps. The steps can include an IESC registering with a SLEE. The SLEE can include an event handler for routing messages between service components executing in the SLEE. The IESC can receive a first event routed by the SLEE. The first event can correspond to an Internet service application which the IESC has registered to receive from the SLEE. Further the first event can be from a protocol stack, a generic service component, a service component, or an external server side program associated with the Internet service application.

The method also can include establishing a communications link between the IESC and an Internet service application. At least one client service instance for communicating with the Internet service application can be instantiated and interact with the Internet service application. A second event can be posted to the SLEE responsive to the interaction between the client service instance and the Internet service application. The second event can correspond to the interaction.

BRIEF DESCRIPTION OF THE DRAWINGS

There are shown in the drawings embodiments of which are presently preferred, it being understood, however, that the invention is not so limited to the precise arrangements and instrumentalities shown.

5 Figure 1 is a schematic representation of an intelligent network architecture configured in accordance with a conventional JAIN implementation known in the prior art.

Figure 2 is a schematic representation of an intelligent network architecture configured in accordance with the inventive arrangements disclosed herein.

10 Figure 3 is a pictorial representation of an exemplary service component for use with the intelligent network architecture of Figure 2.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
21

DETAILED DESCRIPTION OF THE INVENTION

The invention disclosed herein concerns a method and system for providing an interface between an Internet service application and a service logic execution environment (SLEE). In particular, a service component in the SLEE can be pre-
5 configured to communicate with an external server-side program such as a servlet or script. Examples of server side programs can include, but are not limited to, Common Gateway Interface (CGI), Perl scripts, Java™ Server Pages, VB scripts, Active Server Pages™, or other scripting technologies. The pre-configured service component can be referred to as an Internet enabled service component (IESC).

10 Once an external application has gained access to the IESC, the IESC can provide to the external applications access to an event handler in the SLEE through which service components can communicate with one another. Through this mechanism, service components executing in the SLEE also can communicate with the IESC. Thus, because the IESC can be communicatively linked to a server side
15 program executing within a server in a computer communications network such as the Internet, other service components within the SLEE can utilize the IESC to send and receive information over the Internet. Moreover, other service components within the SLEE can be accessed by users and administrators from the Internet.

The combination of server-side programs and the IESC enable various aspects of services and service components to be accessed via the Internet through a common application programming interface (API) provided by the IESC. For example,
20 subscribers can access and modify service attributes. Service attributes can include any subscriber accessible aspect of a service. Similarly, service administrators or super users can access, monitor, and perform administrative functions over the Internet.
25 Administrative functions can include, but are not limited to, any functionality reserved for a system administrator or a super user and not provided to a subscriber. Notably, the IESC can provide this access to subscribers and administrators over the Internet without the aid of external services or systems.

Figure 2 is a schematic illustration of a JAIN-compliant intelligent network configured in accordance with the inventive arrangements. The JAIN-compliant network can include a protocol layer 201, a signaling layer 203, an application layer 205, and a service layer 207. The application layer 205 can host external third party applications. Typical third party applications can suit mass-market demand for services such as virtual private networks (VPNs), inbound services and unified messaging. External third party applications also can include short-lived and niche applications which can be deployed using un-trusted application space deployment technologies such as database lookup interfaces, downloadable mechanisms, and the Parlay API, as are well known in the art. Notably, external applications further can include Internet service applications such as Web based applications 210, implemented at least in part using hypermedia documents, which can be contained within a Web server 209 connected to the Internet and included as part of the World Wide Web.

The Web based applications 210 further can include server side programs for accessing various aspects of service components over the Internet. For example, the server side programs can provide users with the ability to monitor and modify service attributes. As mentioned, service attributes can include any subscriber accessible aspect of services such as call forwarding, call blocking, or messaging. Examples of service attributes can include service information such as telephone numbers to be blocked, times which particular numbers can be blocked, as well as other behavioral aspects of a service such as the type of message to be played to particular calling numbers or the number of rings before the messaging service answers the telephone.

Similarly, server side programs can provide for the administration of service components from the Internet. Administrative functions can include, but are not limited to, any functionality reserved for a system administrator and not provided to a subscriber. For example, such functions can include, but are not limited to monitoring and reporting on various aspects of a service component such as service component usage, the number of users accessing a particular service component, the number of calls being processed by a service component, and monitoring a subscriber's service.

Administrative functions also can include halting the execution of a service component, starting the execution of a service component, and loading and unloading a service component within the SLEE.

5 The service layer 207 can include a SLEE server such as a JSLEE Server 200 which can be configured for compatibility with the JAIN specification. The protocol layer 201 can include one or more protocol stacks which can be configured to interact with the service components 112 and IESCs 114 executing in the JSLEE Server 200 through a signaling layer 203. Notably, the invention is not limited in regard to the number or type of protocol stacks. Rather, JSLEE Server 200 can interact with any
10 protocol stack, for example those protocol stacks configured in accordance with the JAIN specification.

The IESCs 114 can be configured to communicate directly with the Web Server 209 by sending and receiving requests and responses in accordance with the hyper-text transfer protocol (HTTP). Likewise, the Web server 209 can include server side
15 programs configured to communicate with IESCs 114 also in accordance with HTTP. Notwithstanding, the invention is not limited in regard to the manner in which the IESCs and corresponding server side programs communicate. Rather, any known communication technique can suffice including general connection-oriented techniques such as explicit TCP/IP communications.

20 Each IESC 114, like other service components 112 in the JSLEE Server 200, also can be configured with the ability to receive events and post events to the JSLEE Server 200. In this manner, other service components 112 can be registered with the JSLEE Server 200 to receive such events. Those service components 112 can correspondingly post events to the JSLEE Server for which the IESC 114 can be
25 registered to receive. It should be appreciated that the IESC 114 can process events received from the JSLEE Server 200 as well as communications received from the Web Server 209. In this manner, the IESC 114 can function as an interface between the Internet and the JSLEE Server 200.

The JSLEE Server 200 also can include several lifecycle management functions. In particular, the service components 112 and IESCs 114 can be properly loaded within the JSLEE Server 200 for execution. The JSLEE Server 200 can identify configuration and loading parameters associated with each service component 112 and IESC 114 to be loaded. Subsequently, the JSLEE Server 200 can execute the service components 112 and IESC 114 using the identified configuration and loading parameters. Finally, the service components 112 and the IESC 114 can register with an internal event handling component of the JSLEE Server 200 so that events can be transmitted to and from the service components 112 and IESCs 114 executing in the JSLEE Server 200.

In operation, the JSLEE Server 200 can transmit and receive events to and from the protocol stacks in the protocol layer 201. More particularly, the events can be transmitted and received in the event handling component included in the JSLEE Server 200. Likewise, service components 112 and IESCs 114 which are registered with the JSLEE Server can receive protocol stack events directed towards particular ones of the service components 112 and IESCs 114. More specifically, the event handling component can route received events to service components 112 and IESCs 114 which have registered with the JSLEE Server 200 to receive such events. The service components 112 and IESCs 114 further can post protocol stack events to the JSLEE Server 200, as well as other events.

Figure 3 is a schematic diagram illustrating the exemplary IESC 114 of Figure 2. As shown in Figure 3, an IESC 114 can include a service wrapper 406 and one or more client service instances 402. The client service instances 402 can be individually instantiated services. Each client service instance 402 further can register with the event routing bus of the JSLEE Server. Alternatively, the service wrapper 406 can register with the JSLEE Server to receive a particular type of event corresponding to a functionality accessible via the Internet. Accordingly, the service wrapper 406 can receive events from the JSLEE Server and process those events before routing the events to a client service instance 402. The service wrapper 406 further can process information received from the client service instance 402 and format events to be

posted to the JSLEE Server. The service wrapper 406 can instantiate a client service instance 402 for each received event, or series of events comprising a transaction relating to an Internet based service application or functionality. Alternatively, a single client service instance 402 can handle multiple transactions. Thus, each client service instance 402 can interact with the Internet based service application 330. The client service instance 402 can be configured to communicate with server side programs within Web Servers 209 using HTTP requests with reference to the uniform resource locator (URL) of the Web Server 209. Similarly, the Web Servers 209, using server side programs, can communicate with the client service instances 402.

The IESC 114 also can include a deployment descriptor 410, one or more context interfaces 404, and a service interface 408. The deployment descriptor 410 can be a document, such as an XML document, which can describe the proper parameters for initially loading an instance of the IESC 114 in the JSLEE Server. The service interface 408 can publish the interface to the service wrapper 406 to external objects. Notably, the service interface 408 can be included as part of an XML document. Also, the interface to each client service instance 402 can be included as part of a context interface 404, which also can be published as part of an XML document.

The present invention can be realized in hardware, software, or a combination of hardware and software. An IESC for accessing a SLEE from a server side program according to the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system or other apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software can be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein.

The present invention also can be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described

herein, and which when loaded in a computer system is able to carry out these methods. Computer program or application in the present context means any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: a) conversion to another language, code or notation; b) reproduction in a different material form.

5